



**EMBEDDED
LINUX
CONFERENCE**

@



OPEN SOURCE SUMMIT
EUROPE

THE LINUX FOUNDATION

From A Security Expert's Diary

**DOs and DONT's When
Choosing Software for Your
Next Embedded Product**

Marta Rybczynska
Security Lead, Eclipse Oniro
Project for Huawei OSTC

#ossummit @twitterhandle



Day 1

A product

► DEFINING A PRODUCT: HARDWARE SUPPORT

- SOC (System on Chip) vendor support :
 - The SOC itself – is support in Linux mainline ? Device trees ? Quality ?
 - Peripherals – do they have upstream drivers ? Vendor's drivers (which version) ? Quality ?
- Is it supported by distributions/distribution building tools ?
 - Eg. is there a well supported Yocto layer ? Quality : passing yocto-check-layer ?

► DEFINING A PRODUCT: FEATURES

- Network access
 - Which protocols ? Can they use encryption/authentication by default ? Eg. Default to TLS 1.2 or 1.3
 - Cloud functions ? Does it transfer user's data ? How will be the data protected ?
- APIs
 - Which libraries/APIs are must-have ?
- Support : how long ?
 - A typical product consists of hundreds of packages

Day 3

Distribution choice

SAY **NO** TO Do-it-yourself

► DISTRIBUTION CHOICE

- Maintenance

- Do they release security updates ? How frequently ? Are there long-standing security issues ?
- How long is a version supported ? Do they have a Long Time Support version (LTS) ?

- Binary/source ?

- Depending on your needs
- Source : Yocto Project and derivatives (eg. Poky, Oniro), Buildroot
- Binary : Debian, Ubuntu..

► EXAMPLE: YOCTO PROJECT RELEASES

- <https://wiki.yoctoproject.org/wiki/Releases>

Codename	Yocto Project Version	Release Date	Current Version	Support Level	Poky Version	BitBake branch
Langdale	4.1	October 2022		Future - Support for 7 months (until May 2023)	N/A	
Kirkstone	4.0	May 2022	4.0.3 (August 2022)	Long Term Support (minimum Apr. 2024)	N/A	2.0
Dunfell	3.1	April 2020	3.1.19 (August 2022)	Long Term Support (until Apr. 2024)	23.0	1.46

Day 17

Layers bring layers bring layers bring layers

► LAYER DEPENDENCIES: YOCTO PROJECT CASE

- Easy to add layers...
 - Harder to remove what you do not need
- Advice
 - Do you need all those layers ?
 - What is their quality ? Yocto Project Compatible ?
 - Is it maintained ?

► DEPENDENCIES: AN EXAMPLE

- An included layer requests:
`https://github.com/jiazhang0/meta-secure-core`

The screenshot shows the GitHub repository page for `jiazhang0/meta-secure-core`. The repository is public and has 16 watchers, 69 forks, and 67 stars. The navigation bar includes links for Code, Issues (21), Pull requests (1), Actions, Projects, Security, and Insights. The repository has 6 branches and 0 tags. The commit history shows a recent commit by `kkang-wr` and `jiazhang0` titled "cryptfs-tpm2: fix ld warnings with binutils 2.39" 12 days ago, with 481 commits in total. The file list includes `.github`, `meta-efi-secure-boot`, and `meta-encrypted-storage`. The `About` section describes it as an OpenEmbedded layer for secure boot, integrity, and encryption, with tags for security, encryption, uefi, signing-keys, integrity, sgx, tpm, tpm2, ima, efi, secure-boot, and modsign.

jiazhang0 / meta-secure-core Public

Watch 16 Fork 69 Star 67

<> Code Issues 21 Pull requests 1 Actions Projects Security Insights

master 6 branches 0 tags

Go to file Add file Code

kkang-wr and jiazhang0 cryptfs-tpm2: fix ld warnings with binutils 2.39 1a74be5 12 days ago 481 commits

.github	Add .github/CODEOWNERS	5 years ago
meta-efi-secure-boot	meta-secure-core: support kirkstone	3 months ago
meta-encrypted-storage	cryptfs-tpm2: fix ld warnings with binutils 2.39	12 days ago

About

OpenEmbedded layer for the use cases on secure boot, integrity and encryption

security encryption uefi signing-keys integrity sgx tpm tpm2 ima efi secure-boot modsign

Day 27

A little package...

► KNOW WHAT YOU HAVE


- Do you know how many components your project has ?
- Advice ?
 - Check your package list from time to time
 - (When using YP) Use and monitor results of:
 - bitbake -g (dependency graph)
 - cve-check (known security issues in your image)
 - create-spdx (create Software Bill of Materials)

► EXAMPLE: DIGGING DEEPER

- Found libmicrohttpd in the package list
- Dependency chain :
 - Crypto libs→libmicrohttpd→debuginfod (from elfutils)
- Fixing :
 - Need to disable a configuration option in elfutils
 - In YP : via DISTRO_FEATURES

▶ EXAMPLE: USING BEST PRACTICES BADGE (1/2)

- <https://bestpractices.coreinfrastructure.org/en/>



Yocto Project

[Expand panels](#) [Show all details](#) [Hide met & N/A](#)

Projects that follow the best practices below can voluntarily self-certify and show that they've achieved an Open Source Security Foundation (OpenSSF) best practices badge. [Show details](#)

If this is your project, please show your badge status on your project page! The badge status looks like this: [Show details](#)

[openssf best practices](#) [silver](#) Here is how to embed it: [Show details](#)

These are the [passing](#) level criteria. You can also view the [silver](#) or [gold](#) level criteria.

▼ Basics	13/13 ●
▼ Change Control	9/9 ●
▼ Reporting	8/8 ●
▼ Quality	13/13 ●
▼ Security	16/16 ●
▼ Analysis	8/8 ●

▶ EXAMPLE: USING BEST PRACTICES BADGE (2/2)

▼ Basics13/13

▼ Change Control9/9

▲ Reporting8/8

Bug-reporting process

☒ Met

☐ Unmet

☐ ?

The project **MUST** provide a process for users to submit bug reports (e.g., using an issue tracker or a mailing list). (URL required)
[report_process]

https://wiki.yoctoproject.org/wiki/Bug_reporting_and_Information_levels <https://www.yoctoproject.org/tools-resources/bugs>

☒ Met

☐ Unmet

☐ ?

The project **SHOULD** use an issue tracker for tracking individual issues. [report_tracker]

<https://bugzilla.yoctoproject.org/>

☒ Met

☐ Unmet

☐ ?

The project **MUST** acknowledge a majority of bug reports submitted in the last 2-12 months (inclusive); the response need not include a fix. [report_responses]

https://wiki.yoctoproject.org/wiki/Yocto_Project_v2.3_Status

Day 79

Updating now and forever

Day ~~7~~ → 11

Updating now and forever

► UPDATING DEVICES

- Critical
 - Even in best design, there will be bugs (in your code, dependencies, or dependencies of dependencies)
 - Update is usually the ONLY way to solve the issue
- Know your update system scope
 - Firmware
 - Bootloader(s)
 - Kernel
 - Applications

Day ??

What next ?

► MORE STORIES (for part 2)

- Secure boot
- Setting up your network protocols
 - DIY network protocols
 - Disabling/enabling encryption
- Updating Board Support Package (BSP)
- Backporting security patches
- Security Response Team

► TAKE-AWAYS

- Good product security is like healthy living
 - Hard to add later
 - A number of small easy steps
- Evaluating your software is key
 - Do you need it ?
 - Is it good quality ?
 - Does it have security updates ?



EMBEDDED LINUX CONFERENCE



OPEN SOURCE SUMMIT
EUROPE

THE LINUX FOUNDATION